Partitioning sparse matrices for parallel preconditioned iterative methods

Bora Uçar
Emory University, Atlanta, GA

Joint work with Prof C. Aykanat
Bilkent University, Ankara, Turkey

Iterative methods

- Used for solving linear systems $Ax=b$
  - usually A is sparse
- Involves
  - linear vector operations
    - $x = x+\alpha y \quad \therefore\ x_i = x_i + \alpha\, y_i$
  - inner products
    - $\alpha = \langle x,y\rangle \quad \therefore\ \alpha = \Sigma\, x_i \times y_i$
  - sparse matrix-vector multiplies (SpMxV)
    - $y = Ax \quad \therefore\ y_i = \langle A_i,x\rangle$
    - $y = A^T x \quad \therefore\ y_i = \langle A^T{}_i,x\rangle$

while not converged **do**
    computations
    check convergence

Preconditioned iterative methods

- Transform $Ax=b$ to another system that is easier to solve
- Preconditioner is a matrix that does the desired transformation
- Focus: approximate inverse preconditioners
- Right approximate inverse M provides $AM \approx I$
- Instead of solving $Ax=b$, use right preconditioning and solve

$AMy = b$ and then set

$$x = My$$

Parallelizing iterative methods

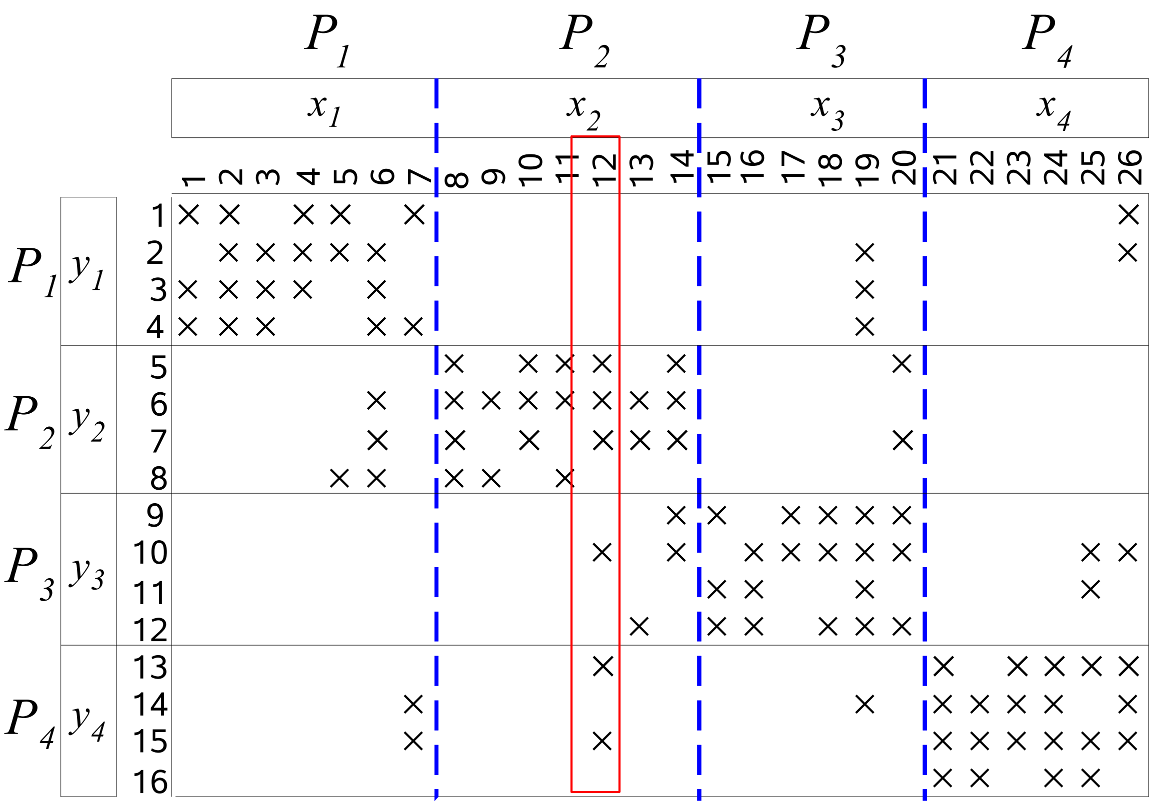
- Avoid communicating vector entries for linear vector operations and inner products
- Inner products require communication
  - regular communication
  - cost remains the same with the increasing problem size
  - there are cost optimal algorithms to perform these communications.
- Efficiently parallelize the SpMxV operations
- Efficiently parallelize the application of the preconditioner

Preconditioned iterative methods

- Applying approximate inverse preconditioners
  - additional SpMxV operations with M
    - never form the matrix AM; perform SpMxVs
- Parallelizing a full step requires efficient SpMxV with A and M
  - partition A and M simultaneously
- What has been done?
  - a bipartite graph model (Hendrickson and Kolda, SISC 00)

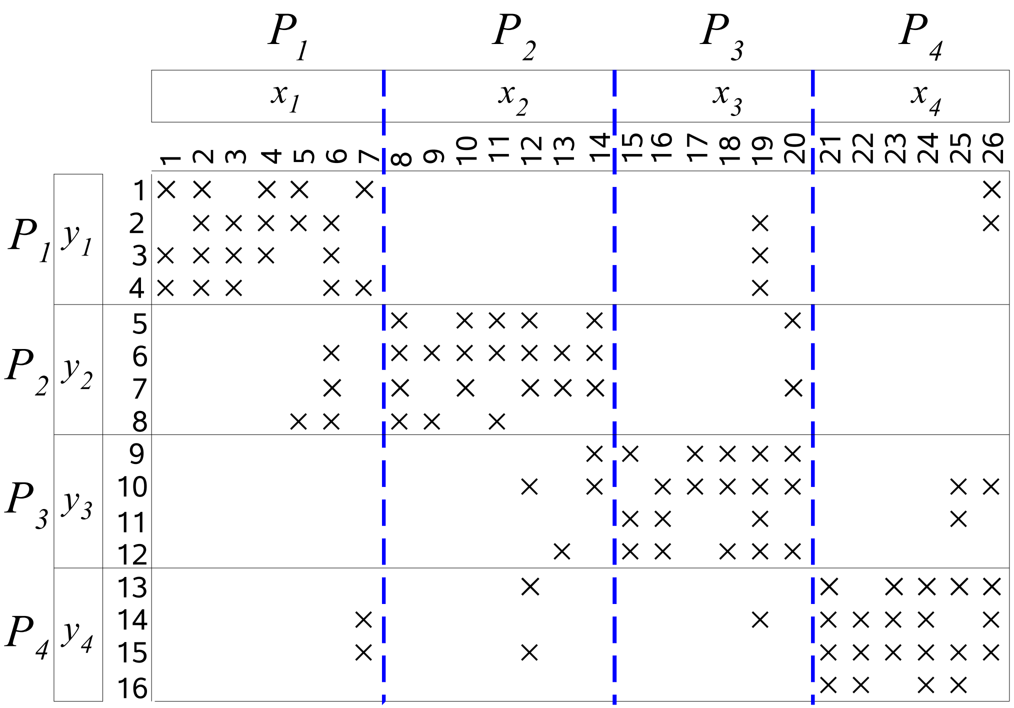
Row-parallel y=Ax

- Rows (and hence y) and x is partitioned



1. Expand *x* vector (sends/receives)
2. Compute with diagonal blocks
3. Receive *x* and compute with off-diagonal blocks

Row-parallel y=Ax



Communication requirements

Total volume:

#nonzero column segments in off diagonal blocks (13)

Total number :

#nonzero off diagonal blocks (9)

Per processor:

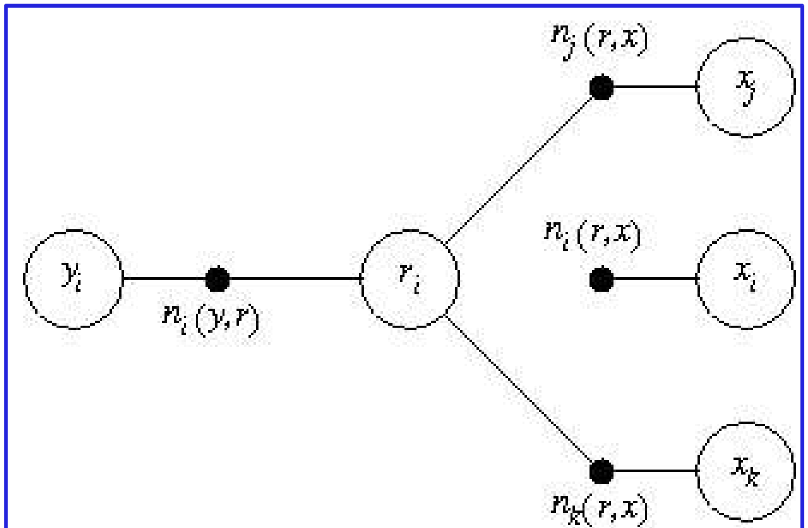
above two confined within a column stripe

Total volume and number of messages addressed previously (Catalyurek and Aykanat, IEEE TPDS 99; U. and Aykanat, SISC 04; Vastenhouw and Bisseling, SIREV 05)

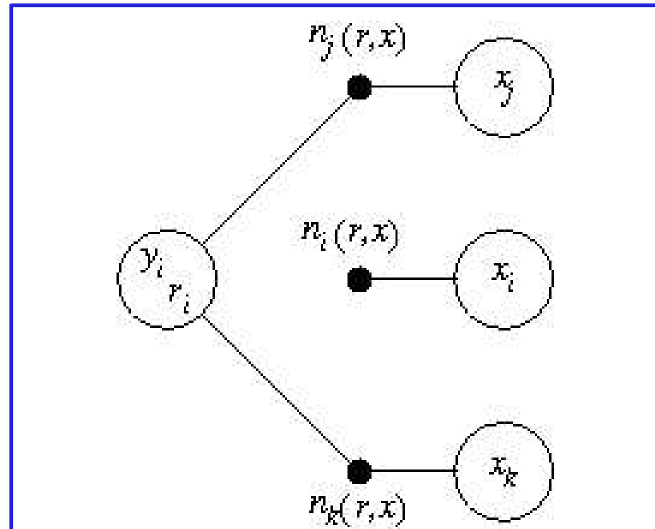
Minimize volume in row-parallel $y=Ax$: Revisiting 1D hypergraph models

- Three entities to partition y, rows of A, & x
  - three types of vertices y_i, r_i & x_j
- y_i is computed by a single r_i
  - connect y_i and r_i (edge, hyperedge)
- x_j is a data source; r_i's where $a_{ij}\neq 0$ need x_j
  - connect x_j and all such r_i (definitely a hyperedge)

Minimize volume in row-parallel y=Ax: Revisiting 1D hypergraph models



General hypergraph model for 1D rowwise partitioning



Combine y_i and r_i: owner computes rule

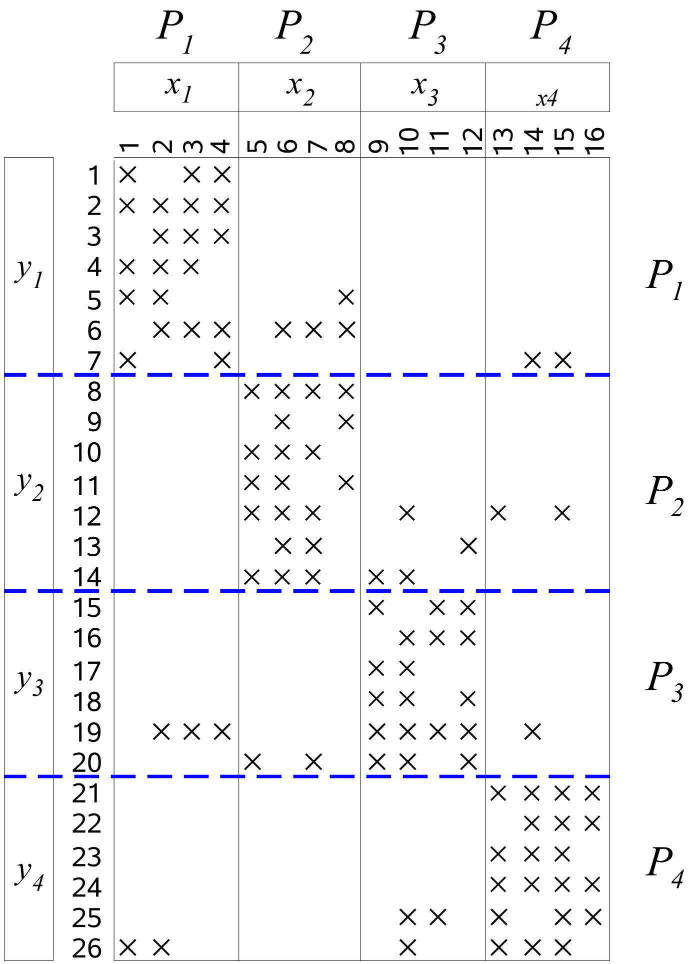
Partition the vertices into *K* parts
(partition the data among *K* processors)

Hypergraph partitioning

- Partition the vertices of a hypergraph into two or more partitions such that:
  - $\sum con(n_i)-1$ is minimized (total volume)

    $con(n_i)$=number of parts connected by hyperedge n_i
  - a balance criterion among the part weights is maintained (load balance)

Column-parallel y=Ax



Communication requirements

Total volume:

#nonzero row segments in off diagonal blocks (13)

Total number :

#nonzero off diagonal blocks (9)

Per processor:

above two confined within a row stripe

Total volume and number of messages addressed previously (Catalyurek and Aykanat, IEEE TPDS 99; U. and Aykanat, SISC 04; Vastenhouw and Bisseling, SIREV 05).

Preconditioned iterative methods

- Linear vector operations and inner product computations are done:
  - all vectors in a single operation have the same partition
- Partition A and M simultaneously
- A blend of dependencies and interactions among matrices and vectors
  - different partitioning requirements in different methods
- Figure out partitioning requirements through analyzing linear vector operations and inner products

Preconditioned BiCG-STAB

$$p^i = r^{i-1} + \beta_{i-1}\left(p^{i-1} - \omega_{i-1} v^{i-1}\right)$$

$$\hat{p} = M p^i$$

$$v^i = A\hat{p}$$

$$s = r^{i-1} - \alpha_i v^i$$

$$\hat{s} = M s$$

$$t = A\hat{s}$$

$$\omega_i = \langle t, s \rangle / \langle t, t \rangle$$

$$x^i = x^{i-1} + \alpha_i p^i + \omega_i s$$

$$r^i = s - \omega_i t$$

p, *r*, *v* should be partitioned conformably

s should be with *r* and *v*

t should be with *s*

x should be with *p* and *s*

Preconditioned BiCG-STAB

$\Rightarrow$ *p*, *r*, *v*, *s*, *t*, and, *x* should be partitioned conformably

- What remains?

$$\hat{p} = Mp^i$$
$$v^i = A\hat{p}$$
$$\hat{s} = Ms$$
$$t = A\hat{s}$$

$\hat{p}$

$\hat{s}$

should be conformal

Columns of M and rows of A should be conformal

Rows of M and columns of A should be conformal

PAQ^T

QMP^T

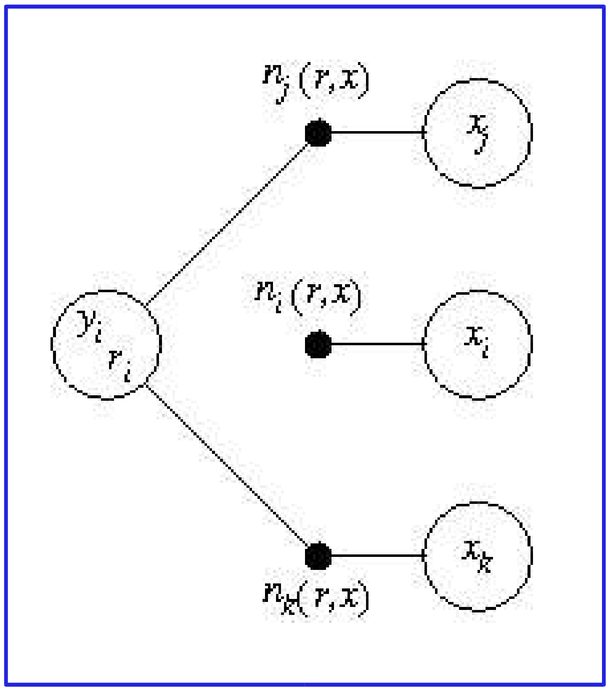
Partitioning requirements

BiCG-STAB	PAQ^TQMP^T
TFQMR	PAP^T and $PM_1M_2P^T$
GMRES	PAP^T and PMP^T
CGNE	PAQ and PMP^T

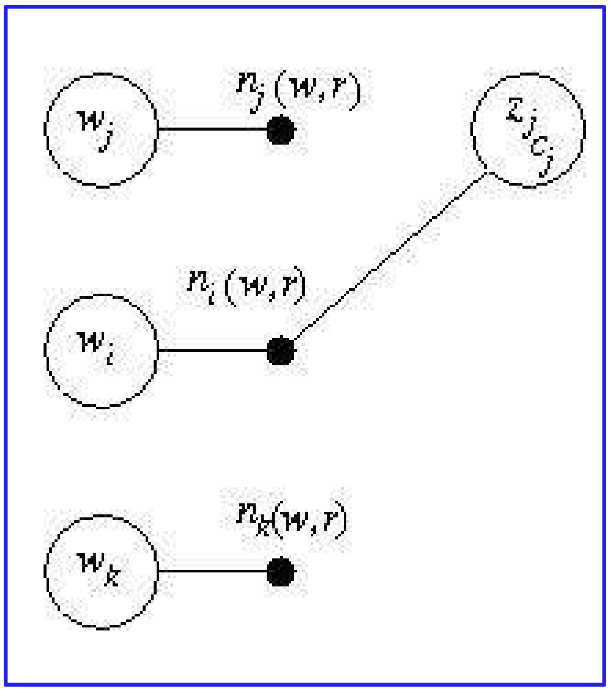
- "and" means there is a synchronization point between SpMxV's
  - Load balance each SpMxV individually

Model for simultaneous partitioning

- We use the previously proposed models
  - define operators to build composite models



Rowwise model (y=Ax)



Col.wise model (w=Mz)

Combining hypergraph models

- Vertex amalgamation: combine vertices of individual hypergraphs, and connect the composite vertex to the hyperedges of the individual vertices

- Vertex weighting: define multiple weights; individual vertex weights are not added up

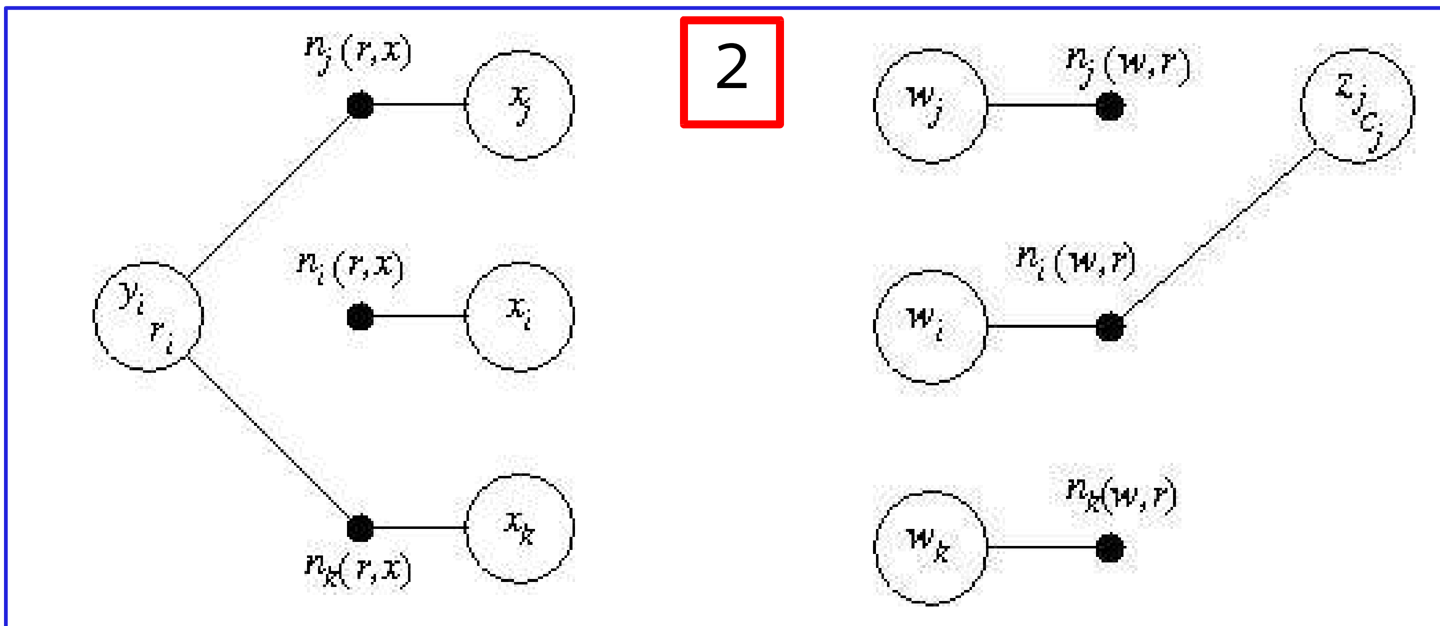
Never amalgamate hyperedges of individual hypergraphs!

Combining guideline

1. Determine partitioning requirements
2. Decide on partitioning dimensions
   - generate rowwise model for the matrices to be partitioned rowwise
   - generate columnwise model for the matrices to be partitioned columnwise
3. Apply vertex operations
   - to impose identical partition on two vertices amalgamate them
   - if the applications of matrices are interleaved with synchronization apply vertex weighting

Combining example

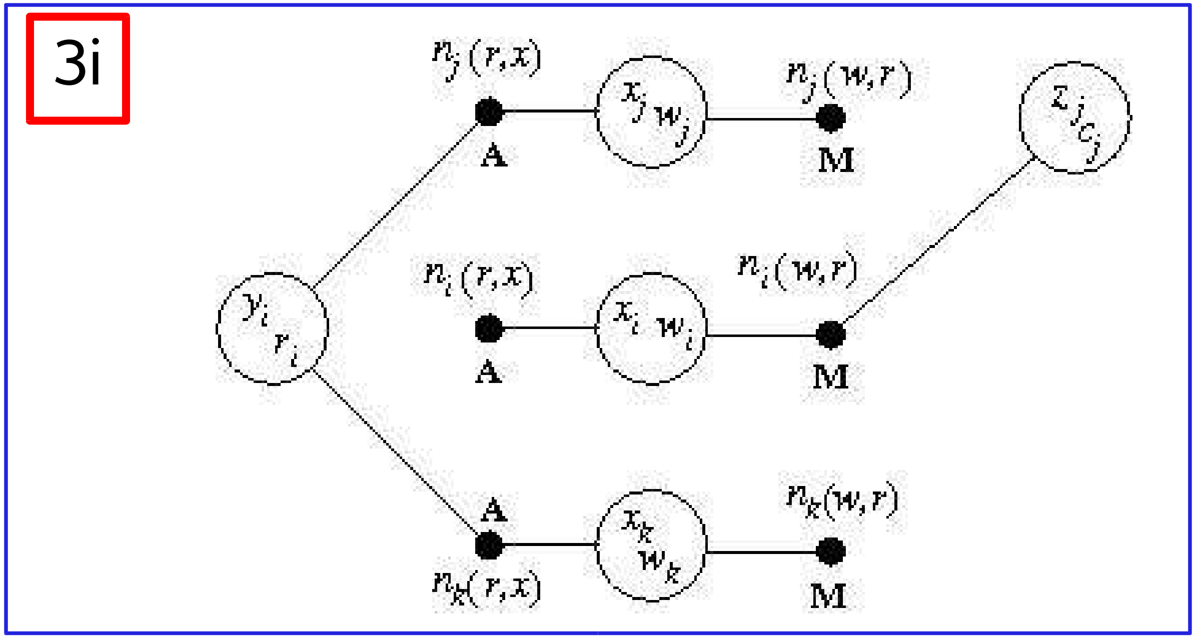
- BiCG-STAB requires PAQ^TQMP^T → 1
- A rowwise ($y=Ax$), M columnwise ($w=Mz$)



Combining example (Cont')

- AQ^TQM: Columns of A and rows of M

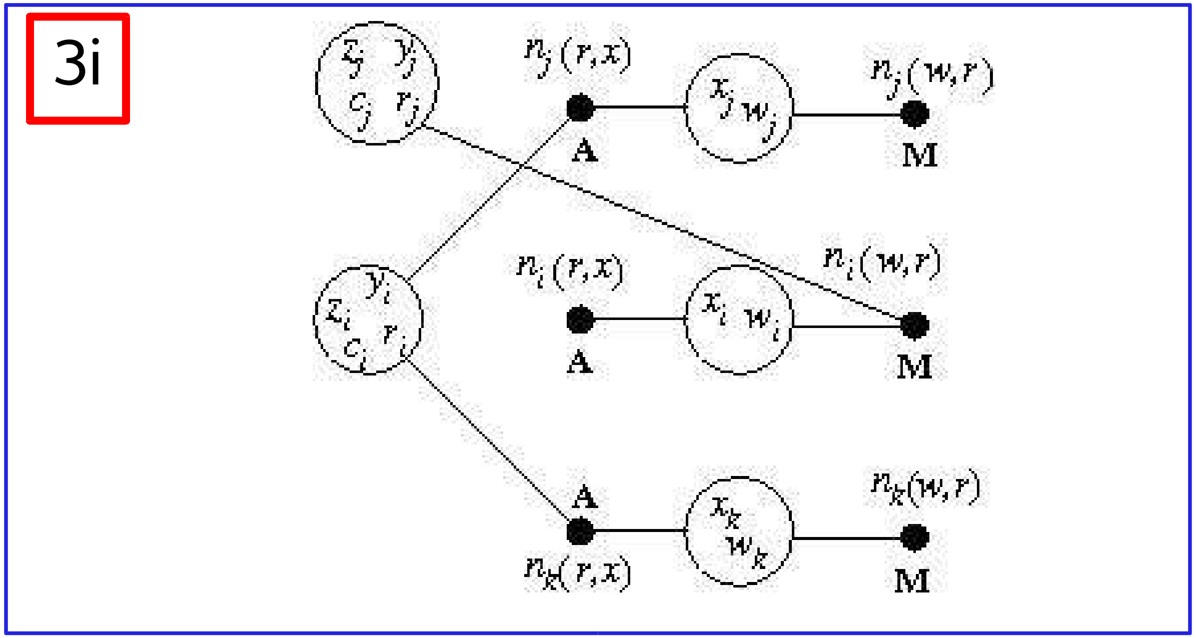
  (y=Ax, w=Mz)



Combining example (Cont')

- PAMP^T: Rows of A and columns of M

  (y=Ax, w=Mz)



Remarks on composite models

- Partitioning the composite hypergraphs
  - balances computational loads of processors
  - minimizes the total communication volume

  in a full step of the preconditioned iterative methods

- Assumption: A and M or their sparsity patterns are available

Experiments: Set up

- Sparse nonsymmetric square matrices from Univ. Florida sparse matrix collection
- SPAI by Grote and Huckle (SISC 97)
- AINV by Benzi and Tůma (SISC 98)
- PaToH by Çatalyürek and Aykanat (TPDS 99)

Experiments: Comparison

With respect to partitioning A and applying the same partition to M (SPAI experiments)

	percent gain in total volume			
	CC		RR	
	32-way	64-way	32-way	64-way
min	7	8	6	8
max	31	34	36	36
average	20	20	20	20

(Ten different matrices)

Experiments: Parallel performance

Parallel BiCGStab speedups (best 5 of the results)
(LAM MPI; 400 MHz Pentium II, 128 Mbyte, Fast ethernet; SPAI)

	Partitioning scheme			
	CR		RC	
	8-procs	16-procs	8-procs	16-procs
stomach	7.1	14.1	7.1	12.7
epb3	7.3	12.4	7.2	11.4
xenon1	6.7	11.2	6.1	9.2
olafu	6.7	10.6	6.0	8.6
cage12	5.9	9.4	4.4	6.2

RC requires multi-constraint formulation

Some other partitioning problems

The principles can be used to parallelize

$$y = (A + B)x$$

$$y = \begin{bmatrix} A \\ B \end{bmatrix} x$$

$$y = \begin{bmatrix} A & B \\ B^T & D \end{bmatrix} x$$

Further information

Thanks: M. Benzi, Ü. V. Çatalyürek, M. Grote, B. Hendrickson, M. Tůma

Ucar and Aykanat, "Partitioning sparse matrices for parallel preconditioned iterative methods", submitted to SISC.

http://www.mathcs.emory.edu/~ubora

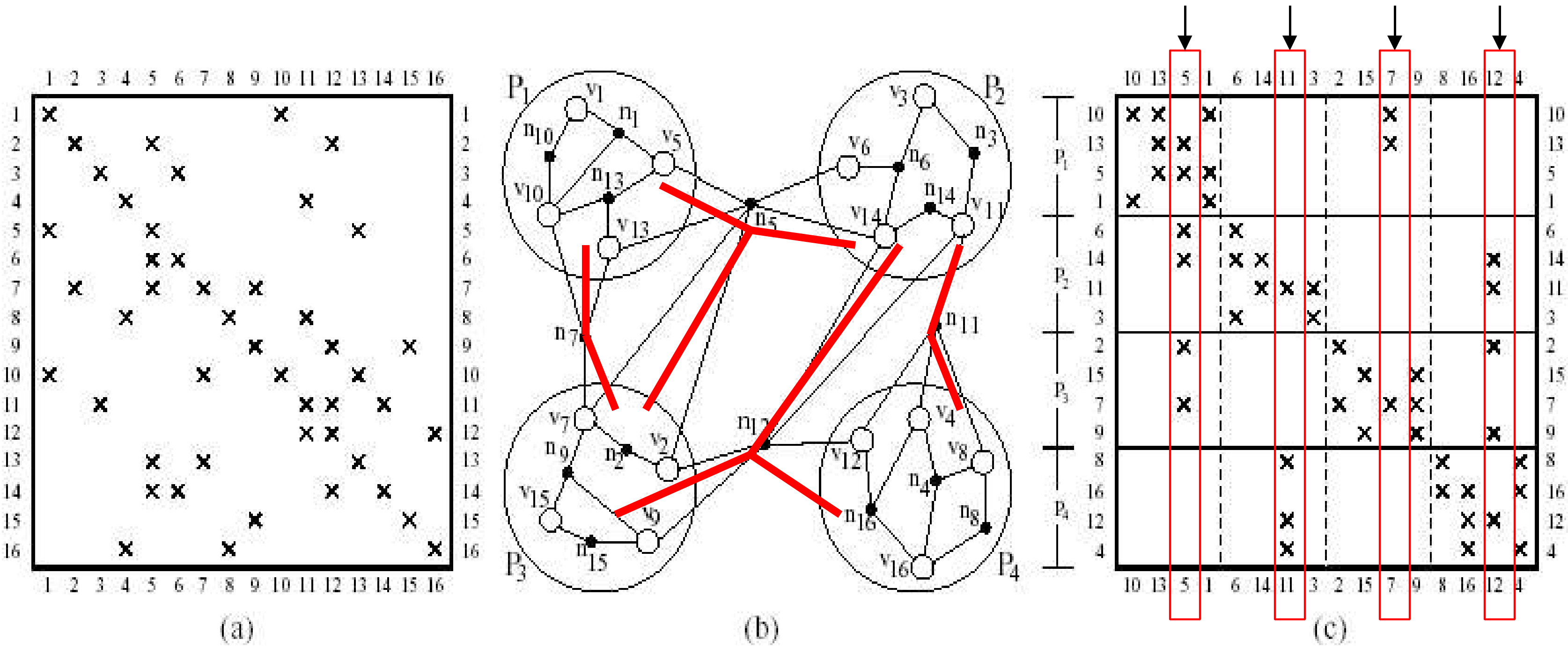
http://www.cs.bilkent.edu.tr/~aykanat

Backups

Hypergraph partitioning

$$\sum con(n_i) - 1 \;=\; 1+2+2+1 = 6$$

Communication volume: 6 words = 2 1 1 2



Matrix properties

Matrix	n	nnz(A)	nnz(M)
big	13209	91465	109088
cage11	39082	559722	424708
cage12	130228	2032536	1444650
epb2	25228	175027	244453
epb3	84617	463625	532851
mark3jac060	27449	170695	276586
olafu	16146	1015156	719873
stomach	213360	3021648	2910283
xenon1	48600	1181120	878143
zhao1	33861	166453	180988

Overlap between sparsity patterns of A and M (SPAI)

	A+M	A\M	M\A	(AnM)/M
Zhao1	234205	67752	53217	0.63
big	147632	56167	38544	0.49
cage11	780776	221054	356068	0.48
cage12	2784199	751663	1339549	0.48
epb2	333794	158767	89341	0.35
epb3	773107	309482	240256	0.42
mark3jac060	397706	227011	121120	0.18
olafu	1357370	342214	637497	0.52
stomach	5182305	2160657	2272022	0.26
xenon1	1520936	339816	642793	0.61

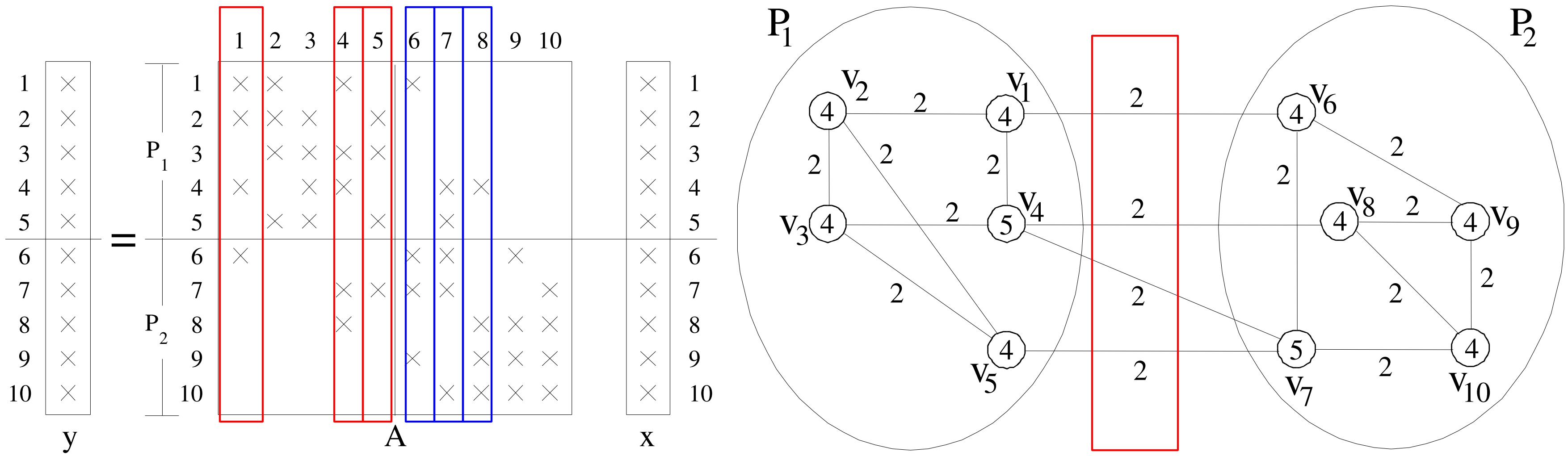
AINV speedups

	K	CRC		RCR	
		Time	S-up	Time	S-up
Zhao1	1	133	1.0	134	1.0
	8	20	6.7	21	6.4
	16	15	8.9	15	8.9
big	1	50	1.0	50	1.0
	8	10	5.0	10	5.0
	16	8	6.3	8	6.3
cage11	1	227	1.0	227	1.0
	8	43	5.3	50	4.5
	16	30	7.6	38	6.0
epb2	1	104	1.0	104	1.0
	8	17	6.1	18	5.8
	16	12	8.7	13	8.0

Graph Partitioning

- Partition the vertices of a graph into two or more partitions such that:
  - weights of the edges among the parts is minimized
  - a balance criterion among the part weights is maintained

Graph Partitioning is Wrong!



edge cut is 8

- P_1 sends 3, P_2 sends 3

total 6 ≠ 8